

**REMARKS**

At the outset, the undersigned thanks the Examiner for the courtesies extended during the telephonic interview conducted on November 2, 2004.

The Final Office Action dated May 17, 2004, has been received and carefully considered. In this response, claims 1, 10, 16, 25, 31, 36, 41, 43, and 45-47 have been amended. Entry of the amendments to claims 1, 10, 16, 25, 31, 36, 41, 43, and 45-47 is respectfully requested. Reconsideration of the outstanding objections/rejections in the present application is also respectfully requested based on the following remarks.

I. THE OBJECTION TO CLAIMS 10-12 AND 25-27

Applicant notes with appreciation the indication on page 17 of the Office Action that claims 10-12 and 25-27 would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Claims 10 and 25 have been amended in this manner, and accordingly these claims should now be allowable. Acknowledgment of same is respectfully requested.

II. THE ANTICIPATION REJECTION OF CLAIMS 31, 33, 35, 36, 38 AND  
40

On page 2 of the Office Action, claims 31, 33, 35, 36, 38 and 40 were rejected under 35 U.S.C. § 102(b) as being anticipated by Wang (Haojin Wang, "Telecommunications Network Management," McGraw-Hill Companies, Inc., July 26, 1999). This rejection is hereby respectfully traversed.

Under 35 U.S.C. § 102, the Patent Office bears the burden of presenting at least a prima facie case of anticipation. In re Sun, 31 USPQ2d 1451, 1453 (Fed. Cir. 1993) (unpublished). Anticipation requires that a prior art reference disclose, either expressly or under the principles of inherency, each and every element of the claimed invention. Id. "In addition, the prior art reference must be enabling." Akzo N.V. v. U.S. International Trade Commission, 808 F.2d 1471, 1479, 1 USPQ2d 1241, 1245 (Fed. Cir. 1986), cert. denied, 482 U.S. 909 (1987). That is, the prior art reference must sufficiently describe the claimed invention so as to have placed the public in possession of it. In re Donohue, 766 F.2d 531, 533, 226 USPQ 619, 621 (Fed. Cir. 1985). "Such possession is effected if one of ordinary skill in the art could have combined the publication's description of the invention with his own knowledge to make the claimed invention." Id.

Although Applicant respectfully disagrees with the pending rejections, Applicant has nonetheless amended the pending independent claims to clarify the invention and to further distinguish the cited references. Specifically, independent claims 1, 16, 31, 36, 41, 43, 45-47 have been amended to recite a feature that is not taught or suggested by any of the applied references, either alone or in combination. Claims 1, 16 and 45, for example, now recite "verifying the existence of all dependency relationship resources of the resource on the system during runtime." Claims 31, 36 and 46 now recite "monitoring the dependency resources during runtime." Claims 41, 43 and 47 now recite "receiving indication of a state change for a first resource during runtime."

Applicant respectfully submits that none of the cited references, alone or in combination, teach or suggest each and every limitation of the pending independent claims, as amended. More specifically, Applicant respectfully submits that none of the cited references, alone or in combination, teach or suggest that the various claimed features take place "during runtime." For example, the Curtis reference expressly relates to "a system, method, program, and data structure for **installing a program** onto a computer including and operating system." The Curtis reference further states that "[t]he program processes

the dependency objects **before installing** the program and determines an operating system command that is capable of determining whether the dependent component indicated in the dependency object is installed in the computer." See Curtis Patent, Col. 3, lines 52-67. In contrast, the claimed systems and methods monitor dependency resources "during runtime," for example. In fact, the Background of the Invention section of the pending application acknowledges the existence of prior art systems and methods that, like Curtis, relate to "specialized **installer** software [having] the ability to check a subset of the dependencies." See Curtis Patent, Page 4, lines 11-14 (emphasis added). Thus, the claimed systems and methods are different from the cited references in several respects, including in the claimed systems and method's ability to monitor dependency resources during runtime. The other cited references are similarly deficient.

Accordingly, because the cited references fail to teach or suggest, alone or in combination, each and every limitation of independent claims 1, 10, 16, 25, 31, 36, 41, 43, and 45-47, claims 1, 10, 16, 25, 31, 36, 41, 43, and 45-47 should be allowable and notice to that effect is courteously solicited.

Claims 2-9, 11-15, 17-24, 26-30, 32-35, 37-40, 42, and 44 are dependent upon independent claims 1, 10, 16, 25, 31, 36, 41,

and 43. Thus, since independent claims 1, 10, 16, 25, 31, 36, 41, and 43 should be allowable as discussed above, claims 2-9, 11-15, 17-24, 26-30, 32-35, 37-40, 42, and 44 should also be allowable at least by virtue of their dependency on one of independent claims 1, 10, 16, 25, 31, 36, 41, and 43. Moreover, these claims recite additional features which are not claimed, disclosed, or even suggested by the cited references taken either alone or in combination. For example, claim 11 recites "wherein a resource can be deployed without satisfying a dependency relationship if the dependency resource is a uses type dependency. Applicant respectfully submits that none of the cited references teach or suggest deployment of a resource without satisfying a dependency relationship if the dependency resource is a uses type dependency.

In view of the foregoing, it is respectfully requested that the aforementioned anticipation rejection of claims 31, 33, 35, 36, 38 and 40 be withdrawn.

### III. THE ANTICIPATION REJECTION OF CLAIMS 41-44 AND 47

On page 4 of the Office Action, claims 41-44 and 47 were rejected under 35 U.S.C. § 102(b) as being anticipated by Svedberg et al. (U.S. Patent No. 5,408,218). This rejection is hereby respectfully traversed.

Applicant respectfully submits that the anticipation rejection of claims 41-44 and 47 has been overcome by the amendments discussed above in connection with the anticipation rejection of claims 31, 33, 35, 36, 38 and 40.

In view of the foregoing, it is respectfully requested that the aforementioned anticipation rejection of claims 41-44 and 47 be withdrawn.

IV. THE OBVIOUSNESS REJECTION OF CLAIMS 1, 3-8, 13, 14, 16, 18-23, 28, 29, 32, 34, 37 AND 39

On page 6 of the Office Action, claims 1, 3-8, 13, 14, 16, 18-23, 28, 29, 32, 34, 37 and 39 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Wang in view of Curtis (U.S. Patent No. 6,442,754). This rejection is hereby respectfully traversed.

Applicant respectfully submits that the obviousness rejection of claims 1, 3-8, 13, 14, 16, 18-23, 28, 29, 32, 34, 37 and 39 has been overcome by the amendments discussed above in connection with the anticipation rejection of claims 31, 33, 35, 36, 38 and 40.

In view of the foregoing, it is respectfully requested that the aforementioned obviousness rejection of claims 1, 3-8, 13, 14, 16, 18-23, 28, 29, 32, 34, 37 and 39 be withdrawn.

V. THE OBVIOUSNESS REJECTION OF CLAIMS 2 AND 17

On page 14 of the Office Action, claims 2 and 17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Wang in view of Curtis and further in view of Deo (U.S. Patent No. 6,594,355). This rejection is hereby respectfully traversed.

Applicant respectfully submits that the obviousness rejection of claims 2 and 17 has been overcome by the amendments discussed above in connection with the anticipation rejection of claims 31, 33, 35, 36, 38 and 40.

In view of the foregoing, it is respectfully requested that the aforementioned obviousness rejection of claims 2 and 17 be withdrawn.

VI. THE OBVIOUSNESS REJECTION OF CLAIMS 9 AND 24

On page 15 of the Office Action, claims 9 and 24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Wang in view of Curtis and further in view of Yue (U.S. Patent No. 6,457,143). This rejection is hereby respectfully traversed.

Applicant respectfully submits that the obviousness rejection of claims 9 and 24 has been overcome by the amendments discussed above in connection with the anticipation rejection of claims 31, 33, 35, 36, 38 and 40.

In view of the foregoing, it is respectfully requested that the aforementioned obviousness rejection of claims 9 and 24 be withdrawn.

VII. THE OBVIOUSNESS REJECTION OF CLAIMS 15 AND 30

On page 15 of the Office Action, claims 15 and 30 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Wang in view of Curtis and further in view of Svedberg. This rejection is hereby respectfully traversed.

Applicant respectfully submits that the obvious rejection of claims 15 and 30 has been overcome by the amendments discussed above in connection with the anticipation rejection of claims 31, 33, 35, 36, 38 and 40.

In view of the foregoing, it is respectfully requested that the aforementioned obviousness rejection of claims 15 and 30 be withdrawn.

VIII. CONCLUSION

In view of the foregoing, it is respectfully submitted that the present application is in condition for allowance, and an early indication of the same is courteously solicited. The Examiner is respectfully requested to contact the undersigned by telephone at the below listed telephone number, in order to expedite resolution of any issues and to expedite passage of the



Patent Application  
Attorney Docket No.: 56130.000042  
Client Reference No.: 12876ROUS01U

present application to issue, if any comments, questions, or suggestions arise in connection with the present application.

To the extent necessary, a petition for an extension of time under 37 CFR § 1.136 is hereby made.

Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 50-0206, and please credit any excess fees to the same deposit account.

Respectfully submitted,

Hunton & Williams LLP

By: 

Thomas E. Anderson

Registration No. 37,063

TEA/OAF

Hunton & Williams LLP  
1900 K Street, N.W.  
Washington, D.C. 20006-1109  
Telephone: (202) 955-1500  
Facsimile: (202) 778-2201

Date: November 8, 2004

BEST AVAILABLE COPY

**APPENDIX A**

1 (**Currently Amended**). A method of dependency management in a component-based system comprising:

defining a resource;

recording an identifier for the resource;

recording resource dependency relationships definitions for the resource;

deploying the resource and the resource dependency relationships of the resource to a system including:

verifying the existence of all dependency relationship resources of the resource on the system during runtime;

transmitting a warning if any of the dependencies of the are unsatisfied;

creating an abstract resource based on a dependency relationship definition of the abstract resource if the abstract resource is not found on the system; and

ending deployment if any dependency relationship is unsatisfied and deployment can not be completed without dependency.

2 (Original). The method of claim 1 wherein defining a resource comprises storing a definition of a resource in a tool to be accessed by a service creation environment ("SCE"), a deployment

tool and a service logic execution environment ("SLEE").

3 (Original). The method of claim 1 wherein recording an identifier to a resource comprises recording an identifier including resource identification, type identification and version.

4 (Original). The method of claim 3 wherein recording an identifier further comprises recording an identifier including scope of the resource.

5 (Original). The method of claim 1 wherein recording dependency information comprises recording associations between the resource identifier and resource identifiers for the dependency relationship resources.

6 (Original). The method of claim 1 wherein recording dependency information comprises automatically recording dependency information.

7 (Original). The method of claim 1 wherein recording dependency information comprises manually recording dependency information through one of: software coding and configuration.

8 (Original). The method of claim 1 wherein recording resource dependency definitions comprises defining dependencies for the resource.

9 (Original). The method of claim 1 wherein recording resource dependency definitions comprises identifying type of dependency for each dependency resource.

10 (**Currently Amended**). A method of dependency management in a component-based system comprising:

defining a resource;

recording an identifier for the resource;

recording resource dependency relationships definitions for the resource, wherein recording resource dependency definitions comprises identifying type of dependency for each dependency resource, ~~The method of claim 9~~ wherein identifying the type of dependency comprises identifying the dependency as one of a resource that is contained by an entity and a resource that is used by the entity, wherein a resource that is contained by the entity is also used by the entity;

deploying the resource and the resource dependency relationships of the resource to a system including:

verifying the existence of all dependency relationship  
resources of the resource on the system;  
transmitting a warning if any of the dependencies of  
the are unsatisfied;  
creating an abstract resource based on a dependency  
relationship definition of the abstract resource if the abstract  
resource is not found on the system; and  
ending deployment if any dependency relationship is unsatisfied  
and deployment can not be completed without dependency.

11 (Previously Presented). The method of claim 10 wherein a resource can be deployed without satisfying a dependency relationship if the dependency resource is a uses type dependency.

12 (Previously Presented). The method of claim 10 wherein identifying the dependency type comprises identifying the dependency type according to the rules:

if entity A uses resource B and resource B uses resource C,  
then A contains C;

if A contains B and B contains C, then A contains C;

if A uses B and B contains C, then A uses C; and

if A contains B and B uses C, then A uses C.

13 (Original). The method of claim 1 wherein deploying the resource comprises using a deployment tool to deploy the resource wherein the deployment tool transmits information regarding dependency relationships to a resource management infrastructure.

14 (Original). The method of claim 1 wherein recording resource dependency relationships definitions for the resource comprises recording dependency relationships to at least one resource pool, the resource pool including a set of homogenous resources used interchangeably on a dynamic basis and allocated to dependent objects as needed.

15 (Original). The method of claim 1 further comprising creating a relationship between the resource and a resource pool manager if the resource will be used interchangeably with other resources in a resource pool, wherein the resource pool manager acts as a proxy for the pooled resources and handles dependency relationships on behalf of the pooled resources.

16 (**Currently Amended**). A system for dependency management in a component-based system comprising:

means for defining a resource;

means for recording an identifier for the resource;

means for recording resource dependency relationships  
definitions for the resource;

means for deploying the resource and the resource  
dependency relationships of the resource to a system including:

means for verifying the existence of all dependency  
relationship resources of the resource on the system during  
runtime;

means for transmitting a warning if any of the  
dependencies of the are unsatisfied;

means for creating an abstract resource based on a  
dependency relationship definition of the abstract resource if  
the abstract resource is not found on the system; and

means for ending deployment if any dependency  
relationship is unsatisfied and deployment can not be completed  
without dependency.

17 (Original). The system of claim 16 wherein the means for  
defining a resource comprises means for storing a definition of  
a resource in a tool to be accessed by a service creation  
environment ("SCE"), a deployment tool and a service logic  
execution environment ("SLEE").

18 (Original). The system of claim 16 wherein the means for recording an identifier to a resource comprises means for recording an identifier including resource identification, type identification and version.

19 (Previously Presented). The system of claim 18 wherein the means for recording an identifier further comprises means for recording an identifier including a scope of the resource.

20 (Original). The system of claim 16 wherein the means for recording dependency information comprises means for recording associations between the resource identifier and resource identifiers for the dependency relationship resources.

21 (Original). The system of claim 16 wherein the means for recording dependency information comprises means for automatically recording dependency information.

22 (Original). The system of claim 16 wherein the means for recording dependency information comprises means for manually recording dependency information through one of: software coding and configuration.



23 (Original). The system of claim 16 wherein the means for recording resource dependency definitions comprises means for defining dependencies for the resource.

24 (Original). The system of claim 16 wherein the means for recording resource dependency definitions comprises means for identifying type of dependency for each dependency resource.

25 (**Currently Amended**). A system for dependency management in a component-based system comprising:

means for defining a resource;

means for recording an identifier for the resource;

means for recording resource dependency relationships definitions for the resource, wherein the means for recording resource dependency definitions comprises means for identifying type of dependency for each dependency resource ~~The system of claim 24~~ wherein the means for identifying the type of dependency comprises means for identifying the dependency as one of a resource that is contained by an entity and a resource that is used by the entity, wherein a resource that is contained by the entity is also used by the entity;

means for deploying the resource and the resource

dependency relationships of the resource to a system including:

means for verifying the existence of all dependency  
relationship resources of the resource on the system;

means for transmitting a warning if any of the  
dependencies of the are unsatisfied;

means for creating an abstract resource based on a  
dependency relationship definition of the abstract resource if  
the abstract resource is not found on the system; and

means for ending deployment if any dependency relationship is  
unsatisfied and deployment can not be completed without  
dependency.

26 (Original). The system of claim 25 wherein a resource can be  
deployed without satisfying a dependency relationship if the  
dependency resource is a uses type dependency.

27 (Previously Presented). The system of claim 25 wherein the  
means for identifying the dependency type comprises means for  
identifying the dependency type according to the rules:

if entity A uses resource B and resource B uses resource C,  
then A contains C;

if A contains B and B contains C, then A contains C;

if A uses B and B contains C, then A uses C; and

if A contains B and B uses C, then A uses C.

28 (Original). The system of claim 16 wherein the means for deploying the resource comprises means for using a deployment tool to deploy the resource wherein the deployment tool transmits information regarding dependency relationships to a resource management infrastructure.

29 (Original). The system of claim 16 wherein the means for recording resource dependency relationships definitions for the resource comprises means for recording dependency relationships to at least one resource pool, the resource pool including a set of homogenous resources used interchangeably on a dynamic basis and allocated to dependent objects as needed.

30 (Original). The system of claim 16 further comprising means for creating a relationship between the resource and a resource pool manager if the resource will be used interchangeably with other resources in a resource pool, wherein the resource pool manager acts as a proxy for the pooled resources and handles dependency relationships on behalf of the pooled resources.

31 (**Currently Amended**). A method of managing dependencies in a component-based system comprising:

performing at least one of a startup and an initialization  
of a resource up to inter-component connection;

determining if the resource has any dependency resources,  
the resource and its dependency resources forming a group of  
resources;

waiting for dependency resources to complete  
initialization;

establishing connections to dependency resources;  
proceeding with the at least one of startup and  
initialization; [[and]]

establishing connections to the resource from the  
dependency resources; and  
monitoring the dependency resources during runtime.

32 (Original). The method of claim 31 further comprising:

receiving indication from the resource that its internal  
resources have been successfully allocated and that the resource  
is waiting for connection;

requesting connection information from an inter-component  
connection manager; and

receiving inter-component connection information from the  
inter-component connection manager.

33 (Original). The method of claim 31 wherein determining if the resource has any dependency resource comprises determining dependency inter-component connection information from inter-component connection information received from an inter-component connection manager.

34 (Original). The method of claim 31 wherein inter-component connection the resources comprises

placing the resource on a ready for inter-component connection list until the dependency resources have been started;

receiving indication from the dependency resource that its internal resources have been successfully allocated and that the dependency resource is waiting for inter-component connection;

requesting inter-component connection information from a inter-component connection manager; and

traversing all entries of inter-component connection information.

35 (Original). The method of claim 31 wherein performing startup of the dependency resources comprises requesting a resource pool manager to assign a dependency resource from the resource pool.

36 (**Currently Amended**). A system for managing dependencies in a component-based system comprising:

means for performing at least one of a startup and an initialization of a resource up to inter-component connection;

means for determining if the resource has any dependency resources, the resource and its dependency resources forming a group of resources;

means for waiting for dependency resources to complete initialization;

means for establishing connections to dependency resources;

means for proceeding with the at least one of startup and initialization; [[and]]

means for establishing connections to the resource from the dependency resources; and

means for monitoring the dependency resources during runtime.

37 (Original). The system of claim 36 further comprising:

means for receiving indication from the resource that its internal resources have been successfully allocated and that the resource is waiting for inter-component connections;

means for requesting inter-component connection information from a inter-component connection manager; and

means for receiving inter-component connection information from the inter-component connection manager.

38 (Original). The system of claim 36 wherein the means for determining if the resource has any dependency resource comprises means for determining dependency inter-component connection information from inter-component connection information received from an inter-component connection manager.

39 (Original). The system of claim 36 wherein the means for performing inter-component connection on the resources comprises means for placing the resource on a ready for inter-component connection list until the dependency resources have been started;

means for receiving indication from the dependency resource that its internal resources have been successfully allocated and that the dependency resource is waiting for inter-component connections;

means for requesting inter-component connection information from a inter-component connection manager; and

means for traversing all entries of inter-component connection information.

40 (Original). The system of claim 36 wherein the means for performing startup of the dependency resources comprises means for requesting a resource pool manager to assign a dependency resource from the resource pool.

41 (**Currently Amended**). A method of managing dependencies in a component-based system comprising:

receiving indication of a state change for a first resource during runtime;

transmitting the indication of the state change of the first resource to a second resource dependent on the first resource; and

receiving indication of a state change of the second resource.

42 (Original). The method of claim 41 wherein receiving indication of the state change of the first resource comprises receiving indication of the state change from a managed object view of the first resource, transmitting the indication of state change to the second resource comprises transmitting the indication to a managed object view of the second resource and receiving indication of the state change of the second resource comprises receiving the indication of state change from the



managed object view of the second resource.

43 (**Currently Amended**). A system for managing dependencies in a component-based system comprising:

means for receiving indication of a state change for a first resource during runtime;

means for transmitting the indication of the state change of the first resource to a second resource dependent on the first resource; and

means for receiving indication of a state change of the second resource.

44 (Original). The system of claim 43 wherein the means for receiving indication of the state change of the first resource comprises means for receiving indication of the state change from a managed object view of the first resource, the means for transmitting the indication of state change to the second resource comprises means for transmitting the indication to a managed object view of the second resource and the means for receiving indication of the state change of the second resource comprises means for receiving the indication of state change from the managed object view of the second resource.

45 (**Currently Amended**). A computer readable medium, the computer readable medium storing computer readable code executable to perform a method for managing a component-based system comprising:

- defining a resource;
- recording an identifier for the resource;
- recording resource dependency relationships definitions for the resource;

- deploying the resource and the resource dependency relationships of the resource to a system including:

- verifying the existence of all dependency relationship resources of the resource on the system during runtime;

- transmitting a warning if any of the dependencies of the are unsatisfied;

- creating an abstract resource based on a dependency relationship definition of the abstract resource if the abstract resource is not found on the system; and

- ending deployment if any dependency relationship is unsatisfied and deployment can not be completed without dependency.

46 (**Currently Amended**). A computer readable medium, the computer readable medium storing computer readable code

executable to perform a method for managing a component-based system comprising:

performing at least one of a startup and an initialization of a resource up to inter-component connection;

determining if the resource has any dependency resources, the resource and its dependency resources forming a group of resources;

waiting for dependency resources to complete initialization;

establishing connections to dependency resources;

proceeding with the at least one of startup and initialization; [[and]]

establishing connections to the resource from the dependency resources; and

monitoring the dependency resources during runtime.

47 (**Currently Amended**). A computer readable medium, the computer readable medium storing computer readable code executable to perform a method for managing a component-based system comprising:

receiving indication of a state change for a first resource during runtime;

transmitting the indication of the state change of the

first resource to a second resource dependent on the first  
resource; and

receiving indication of a state change of the second  
resource.